

WHAT IS CLAIMED IS:

1. A method of managing membership of members in a cluster, comprising:
providing a domain for each member of a group, wherein the domain indicates all members of the cluster with a membership to the group; and
providing a set of interfaces configured to be invoked to manage the membership to the group, wherein at least one interface, when invoked by a request of a requestor, causes each member of the group to access its respective copy of the domain to determine whether the requestor is indicated its respective copy of the domain.
2. The method of claim 1, wherein the request is to join the requestor to the group.
3. The method of claim 1, wherein providing the set of interfaces comprises providing a Join_Member interface which is invoked when an inactive member requests to join the group, and wherein the Join_Member interface, when invoked, causes an active member currently in the group to access its copy of the domain to determine whether the inactive member has membership with the group.
4. The method of claim 1, wherein providing the set of interfaces comprises providing a first interface invoked by a request to add a potential member to the group, a second interface invoked by a request to join an inactive member to the group, and a third interface invoked by a request to remove the a member from the group.
5. The method of claim 1, wherein the domain is a unique persistent object in the cluster.
6. The method of claim 1, wherein the members are jobs running on nodes of the cluster.
7. The method of claim 6, wherein the members are differentiated by unique names of a respective node on which they are running.
8. A method of managing membership of jobs in a cluster, the method comprising:

- (i) upon receiving a request to create a group comprising at least two jobs:
creating, on a respective node on which each of the at least two jobs is
running, a list indicating each of the at least two jobs; and
- (ii) upon receiving a request to join the group from a requesting member job
having membership to the group:
accessing each list of each job of the group to determine whether the
requesting member job is included in each list.
9. The method of claim 8, further comprising:
determining that the requesting member job is included in at least one list; and
joining the requesting member job to the group.
10. The method of claim 8, further comprising, upon receiving a request to leave the
group from a requesting member job having membership to the group:
updating each list of each job of the group to remove the requesting member job
from the list.
11. The method of claim 8, further comprising, upon receiving a request to add a new
job to the group:
for each current member of the group, updating a respective list to include
the new job; and
for the new node, replicating the list to the new job; and
12. A computer system, comprising a first plurality of nodes, each node comprising:
a processor configured to execute at least one job; and
a memory device containing a copy of a first list; wherein each copy of the first list
indicates jobs with a membership to a first group and wherein each job is configured to
access its respective copy of the first list to determine whether a requesting job of
another node may be joined to the first group.

13. The system of claim 12, further comprising a plurality of interfaces configured for adding jobs to the first group, removing jobs from the first group, and joining returning member jobs to the first group.

14. The system of claim 12, wherein each job is configured to update its respective copy of the first list to include added members.

15. The system of claim 12, wherein each job is configured to update its respective copy of the first list to remove dropped members.

16. The system of claim 12, wherein the requesting job is joined to the first group when the first list contains a reference to a node on which the requesting job is running.

17. The system of claim 12, further comprising:
a second plurality of nodes; and
a copy of a second list stored on each of the second plurality of nodes and associated with a job executing on the each of the second plurality of nodes; wherein each copy of the second list indicates a membership to a second group.

18. The system of claim 17, wherein the copies of the first list and the copies of the second list are each unique on the system.

19. A memory of a node in a cluster, the memory containing at least a data structure, the data structure comprising a list defining membership to a group; wherein the list is replicated to each job having membership to the group and wherein each list is accessed upon each request from a requesting member job to join the group, wherein the request is granted if the other jobs of the group determine that the requesting member job is indicated in each respective list of the other jobs.

20. The memory of claim 19, wherein each list is updated to include a new job upon each granted request from the new job to be added to the group.

21. The memory of claim 19, wherein each list is updated to remove a leaving member job upon each granted request from the leaving member job to be removed to the group.

22. The memory of claim 19, wherein the list is unique within the cluster.

23. The memory of claim 19, further comprising a plurality of interfaces comprising a first interface invoked by a request to add a new job to the group, a second interface invoked by a request to rejoin a requesting job to the group, a third interface invoked by a request to remove the requesting member job from the group.